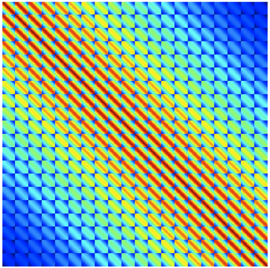
Quasi-Monte Carlo methods for computing flow in random porous media

Dirk Nuyens[a,(c)]

Joint work with I. G. Graham[b], F. Y. Kuo[c], R. Scheichl[b], and I. H. Sloan[c]

[a]Department of Computer Science, K.U.Leuven, Belgium
[b]Department of Mathematical Sciences, University of Bath, UK
[c]School of Mathematics & Statistics, University of NSW, Australia



ICCAM 2010, K.U.Leuven, Belgium, 5–9 July 2010

Outline

1. Introduction
2. Simulating a random field
3. Circulant embedding
4. Numerical results
5. Conclusions

Sketch of the problem: flow through random media

- "Random porous media"
- Darcy's law + mass conservation:

$$\vec{q} + k\,\vec{\nabla} p = \vec{0},$$
$$\vec{\nabla} \cdot \vec{q} = 0$$

  seek velocity $\vec{q}$ and pressure p
- Finite element method
  $\Rightarrow m \times m$ grid
- Permeability k
  $\Rightarrow$ *random field*
- Solve N times

Sketch of the problem: flow through random media

- “Random porous media”
- Darcy’s law + mass conservation:

$$\vec{q} + k\,\vec{\nabla} p = \vec{0},$$
$$\vec{\nabla} \cdot \vec{q} = 0$$

  seek velocity $\vec{q}$ and pressure p
- Finite element method
  $\Rightarrow m \times m$ grid
- Permeability k
  $\Rightarrow$ *random field*
- Solve N times

Sketch of the problem: flow through random media

- "Random porous media"
- Darcy's law + mass conservation:

$$\vec{q} + k\,\vec{\nabla} p = \vec{0},$$
$$\vec{\nabla} \cdot \vec{q} = 0$$

  seek velocity $\vec{q}$ and pressure p
- Finite element method
  $\Rightarrow m \times m$ grid
- Permeability k
  $\Rightarrow$ *random field*
- Solve N times

Sketch of the problem: flow through random media

- "Random porous media"
- Darcy's law + mass conservation:

$$\vec{q} + k\,\vec{\nabla} p = \vec{0},$$
$$\vec{\nabla} \cdot \vec{q} = 0$$

seek velocity $\vec{q}$ and pressure p

- Finite element method
$\Rightarrow m \times m$ grid
- Permeability k
$\Rightarrow$ *random field*
- Solve N times

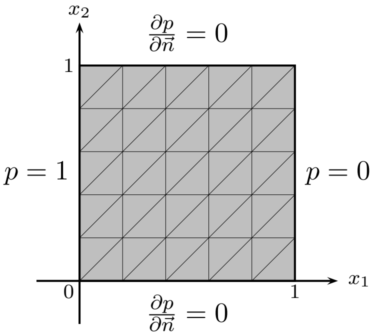
Sketch of the problem: flow through random media

- "Random porous media"
- Darcy's law + mass conservation:

$$\vec{q} + k\,\vec{\nabla} p = \vec{0},$$
$$\vec{\nabla} \cdot \vec{q} = 0$$

seek velocity $\vec{q}$ and pressure p

- Finite element method
$\Rightarrow m \times m$ grid
- Permeability k
$\Rightarrow$ *random field*
- Solve N times

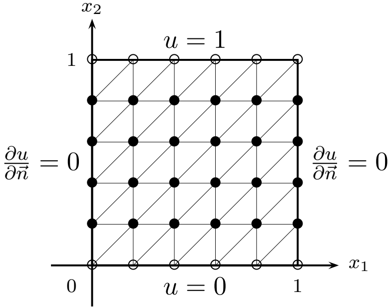


Sketch of the problem: flow through random media

- "Random porous media"
- Darcy's law + mass conservation:

$$\vec{q} + k\,\vec{\nabla} p = \vec{0},$$
$$\vec{\nabla} \cdot \vec{q} = 0$$

  seek velocity $\vec{q}$ and pressure p
- Finite element method
  $\Rightarrow m \times m$ grid
- Permeability k
  $\Rightarrow$ *random field*
- Solve N times

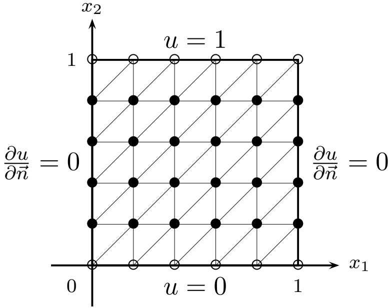


Sketch of the problem: flow through random media

- "Random porous media"
- Darcy's law + mass conservation:

$$\vec{q} + k\,\vec{\nabla} p = \vec{0}, \qquad \vec{\nabla}\cdot\vec{q} = 0$$

  seek velocity $\vec{q}$ and pressure p
- Finite element method
  $\Rightarrow m \times m$ grid
- Permeability k
  $\Rightarrow$ *random field*
- Solve N times

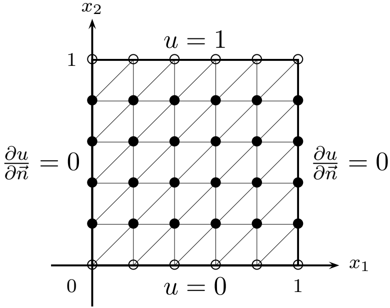


Sketch of the problem: flow through random media

- "Random porous media"
- Darcy's law + mass conservation:

$$\vec{q} + k\,\vec{\nabla} p = \vec{0},$$
$$\vec{\nabla} \cdot \vec{q} = 0$$

  seek velocity $\vec{q}$ and pressure p
- Finite element method
  $\Rightarrow m \times m$ grid
- Permeability k
  $\Rightarrow$ *random field*
- Solve N times



$\rightarrow$ study nonlinear functionals of random field by simulation

Technology and aim

Technology:

- Quasi-Monte Carlo for simulating the random field
- As a d-dimensional integral
  Ultra-high dimensionality $(d > N)$; here $d \approx 10^6$
- Circulant embedding, real factorization
- Mixed finite elements
- Divergence-free reduction
- Algebraic multigrid

Aim:

- Accelerate Monte Carlo
- Handle rough fields
  where Karhunen-Loève does not converge fast enough

Point sets to choose from

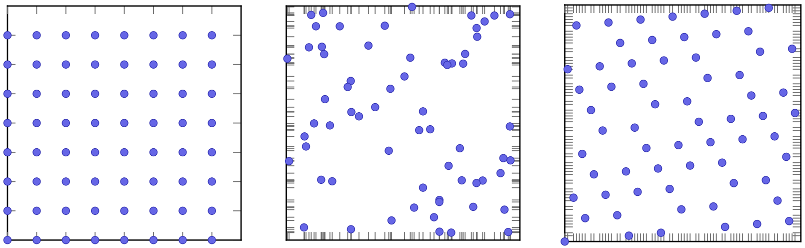


Figure: Three point sets with each 64 samples in the unit square.

1. The product left-rectangle rule. → Classical product rule
   Note: taking 2 points per dimensions in 100 dimensions requires quintillion $2^{100} = 1267650600228229401496703205376$ points…
2. Pseudo-random numbers. → Monte Carlo
3. *Low-discrepancy points.* → Quasi-Monte Carlo

Point sets to choose from

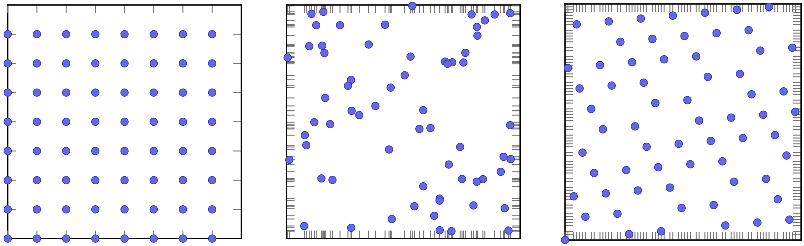


Figure: Three point sets with each 64 samples in the unit square.

1. The product left-rectangle rule. → Classical product rule
2. Pseudo-random numbers. → Monte Carlo
   (Fig: mt19937, the Mersenne Twister, with default initial state.)
3. *Low-discrepancy points.* → Quasi-Monte Carlo

Point sets to choose from

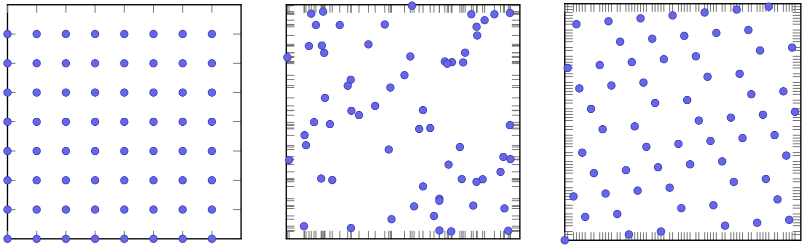
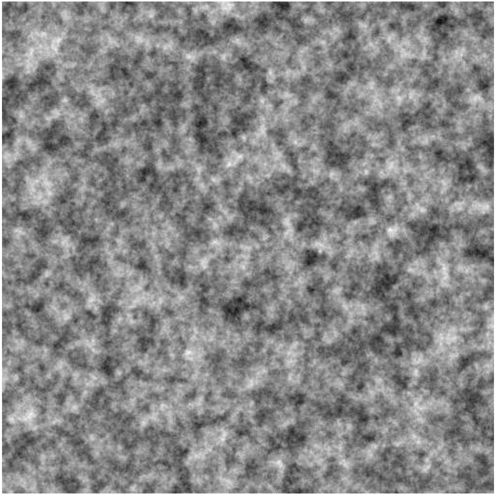


Figure: Three point sets with each 64 samples in the unit square.

1. The product left-rectangle rule. → Classical product rule
2. Pseudo-random numbers. → Monte Carlo
3. *Low-discrepancy points.* → Quasi-Monte Carlo
   (Fig: A good lattice sequence in base 3.)

Simulating a random field



The random field model

Permeability k is a log-normal random field

$$k(\vec{x};\omega) = \exp(Z(\vec{x};\omega)),$$

with Z a zero mean Gaussian random field with covariance

$$r(\vec{x},\vec{y}) = \sigma^2 \exp(-\|\vec{x}-\vec{y}\|/\lambda)$$

with variance σ^2 and length scale λ.

- λ models the effect of distance interaction
- σ^2 models the difference between mins and maxs

More difficult: $\sigma^2 \nearrow$ and/or $\lambda \searrow$.

Note: need discretization $10h \lesssim \lambda^{-1}$.

Two methods of simulating random field

IDEA: Simulate k only at the nodal points of the FE grid.

Two methods of simulation:

1. Karhunen-Loève (KL) expansion: *Not!*

$$Z(\vec{x};\omega) = \sum_{k=1}^{\infty} \sqrt{\mu_k}\,\phi_k(\vec{x})\,Y_k(\omega) \approx \sum_{k=1}^{K} \sqrt{\mu_k}\,\phi_k(\vec{x})\,Y_k(\omega)$$

$$\mathbb{E}(\mathcal{G}(Z)) \overset{\text{KL}}{\approx} \mathbb{E}(\mathcal{G}(\widetilde{\mathbf{Z}})) \overset{\text{FE}}{\approx} \mathbb{E}(\mathcal{G}_h(\widetilde{\mathbf{Z}})) := \int_{[0,1]^K} \mathcal{G}_h\left(\Psi\boldsymbol{\Phi}_K^{-1}(\mathbf{u})\right) \mathbf{du} \approx \cdots$$

2. Discrete simulation → *circulant embedding*: *Yes!*

$$\mathbb{E}(\mathcal{G}(Z)) \overset{\text{FE}}{\approx} \mathbb{E}(\mathcal{G}_h(\mathbf{Z})) \overset{\text{QMC}}{\approx} \frac{1}{N}\sum_{n=1}^{N} \mathcal{G}_h\left(\Theta\boldsymbol{\Phi}_d^{-1}(\mathbf{u}^{(n)})\right)$$

Two methods of simulating random field

IDEA: Simulate k only at the nodal points of the FE grid.

Two methods of simulation:

1. Karhunen-Loève (KL) expansion: *Not!*

$$Z(\vec{x};\omega) = \sum_{k=1}^{\infty} \sqrt{\mu_k}\,\phi_k(\vec{x})\,Y_k(\omega) \approx \sum_{k=1}^{K} \sqrt{\mu_k}\,\phi_k(\vec{x})\,Y_k(\omega)$$

$$\mathbb{E}(\mathcal{G}(Z)) \overset{\text{KL}}{\approx} \mathbb{E}(\mathcal{G}(\widetilde{\mathbf{Z}})) \overset{\text{FE}}{\approx} \mathbb{E}(\mathcal{G}_h(\widetilde{\mathbf{Z}})) := \int_{[0,1]^K} \mathcal{G}_h\big(\Psi\mathbf{\Phi}_K^{-1}(\mathbf{u})\big)\,\mathbf{du} \approx \cdots$$

2. Discrete simulation → *circulant embedding*: *Yes!*

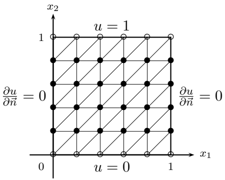
$$\mathbb{E}(\mathcal{G}(Z)) \overset{\text{FE}}{\approx} \mathbb{E}(\mathcal{G}_h(\mathbf{Z})) \overset{\text{QMC}}{\approx} \frac{1}{N}\sum_{n=1}^{N} \mathcal{G}_h\big(\Theta\mathbf{\Phi}_d^{-1}(\mathbf{u}^{(n)})\big)$$

Circulant embedding

We have a Toeplitz structure in the covariance matrix as

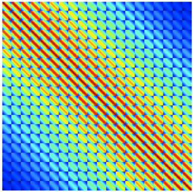
$$r(\vec{x}, \vec{y}) = \sigma^2 \exp(-\|\vec{x} - \vec{y}\|/\lambda)$$

on a regular 2D grid gives a symmetric block Toeplitz matrix



$m \times m$

$\mapsto$



$m^2 \times m^2$

Generate correlated normal variates

To generate $\mathbf{Z} \sim N_d(\mathbf{0}, \Sigma)$:

$$\Sigma = AA^\top$$

then given $\mathbf{Y} \sim N_d(\mathbf{0}, I_d)$

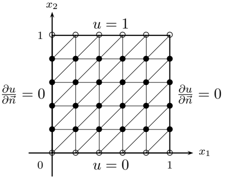
$$\mathbf{Z} = A\mathbf{Y} \sim N_d(\mathbf{0}, \Sigma)$$

However, interest in rough fields:
$\to$ h should be small enough
$\to$ stochastic dimension d will be too large for Cholesky

Circulant embedding

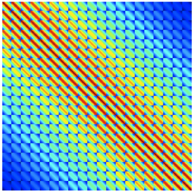
Standard trick: embed Toeplitz structure in circulant structure

Symmetric block Toeplitz becomes block circulant



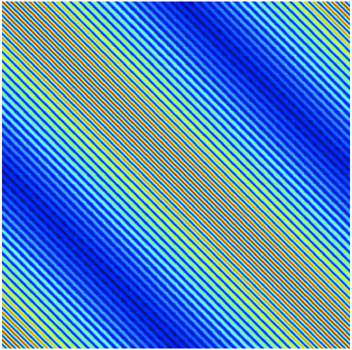
$m \times m$

$\mapsto$



$m^2 \times m^2$

$\mapsto$

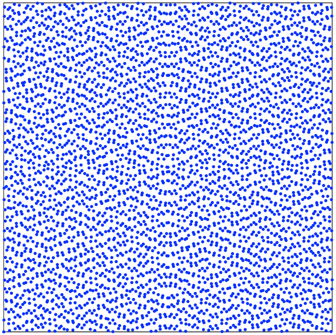


$\gtrsim (2m)^2 \times (2m)^2$

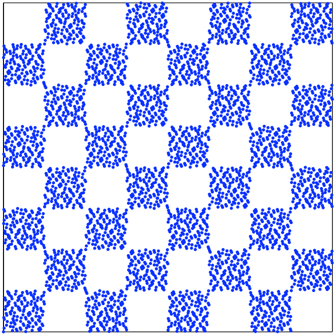
Needs to be positive definite, could be much larger: “padding”

Quasi-Monte Carlo: Opening dimensions are better

Quasi-Monte Carlo points degrade in quality for increasing number of dimensions



dimension 2 versus 9



dimension 20 versus 29

A real factorization

If F is the 2D Fourier matrix

$$F = \Big(\exp(2\pi\mathrm{i}\ p_1 q_1/d_1)\exp(2\pi\mathrm{i}\ p_2 q_2/d_2)\Big)_{\substack{p_1,q_1=0,\ldots,d_1-1\\ p_2,q_2=0,\ldots,d_2-1}}$$

then for $\Lambda = \mathrm{diag}(F\,\mathbf{c}_1)$

$$\begin{aligned} C &= F^{\mathsf{H}}\,\Lambda\,F && (1)\\ &= G\,\Lambda\,G^{\top} && (2)\end{aligned}$$

with $G = \mathrm{Re}(F) + \mathrm{Im}(F)$

$\rightarrow$ Monte Carlo: complex factorization (1): 2 in 2 out (i.i.d.)
  $\rightarrow$ does not work for QMC!
    $\rightarrow$ Quasi-Monte Carlo: (2) is real

Modifications for quasi-Monte Carlo

- Limit nominal dimension by using real factorization
- Order QMC dimensions by magnitude of eigenvalues

A note on speed: Also for QMC we can get “two for the price of one” with minor post processing of the FFT; in fact the cost is approximately the same

Numerical results

We had two aims:

- Accelerate Monte Carlo
- Handle rough fields

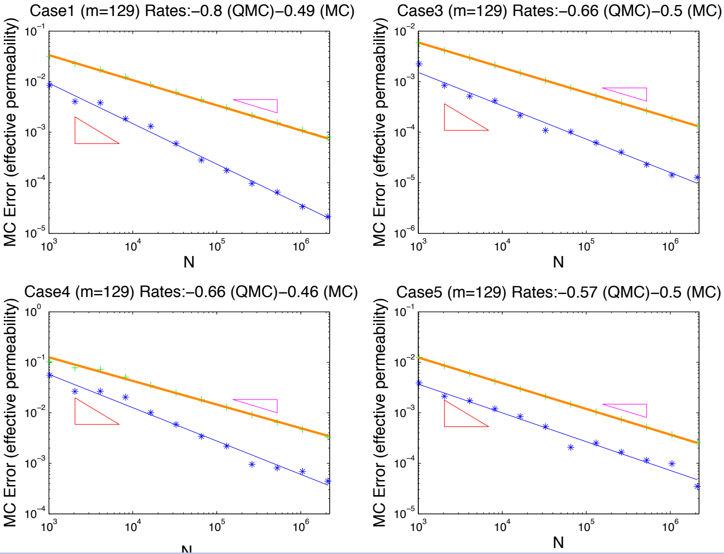
We achieved both!

Case 1	Case 2	Case 3	Case 4	Case 5
$\sigma^2 = 1,\ \lambda = 1$	$\sigma^2 = 1,\ \lambda = 0.3$	$\sigma^2 = 1,\ \lambda = 0.1$	$\sigma^2 = 3,\ \lambda = 1$	$\sigma^2 = 3,\ \lambda = 0.1$

Difficulty goes approximately from left to right

Timing results like: 1 hour instead of 5 days, 28 minutes versus 28 days, …

Calculating effective permeability



Conclusions

- Quasi-Monte Carlo did better than Monte Carlo in all tested cases
- Speed ups between 4 and 200
- Solving hard (rough) problems
- No error bounds (yet?)
- Extensible to 3D

Note: software for generating QMC points?
$\rightarrow$ http://people.cs.kuleuven.be/dirk.nuyens/qmc-generators/
$\rightarrow$ Ask me!